An estimation of distribution algorithm with adaptive Gibbs sampling for unconstrained global optimization

Jonás Velasco, Mario A. Saucedo-Espinosa, Hugo Jair Escalante,
Karlo Mendoza, César Emilio Villarreal-Rodríguez,
Óscar L. Chacón-Mondragón, Adrián Rodríguez, Arturo Berrones
Posgrado en Ingeniería de Sistemas, FIME, UANL
AP 126, Cd. Universitaria, San Nicolás de los Garza, NL 66450, México
jonas.velascoal@uanl.edu.mx

July 12, 2011

Abstract

In this paper is proposed a new heuristic approach belonging to the field of evolutionary Estimation of Distribution Algorithms (EDAs). EDAs builds a probability model and a set of solutions is sampled from the model which characterizes the distribution of such solutions. The main framework of the proposed method is an estimation of distribution algorithm, in which an adaptive Gibbs sampling is used to generate new promising solutions and, in combination with a local search strategy, it improves the individual solutions produced in each iteration. The Estimation of Distribution Algorithm with Adaptive Gibbs Sampling we are proposing in this paper is called AGEDA. We experimentally evaluate and compare this algorithm against two deterministic procedures and several stochastic methods in three well known test problems for unconstrained global optimization. It is empirically shown that our heuristic is robust in problems that involve three central aspects that mainly determine the difficulty of global optimization problems, namely high-dimensionality, multi-modality and non-smoothness.

Key words: Estimation of distribution algorithms; Evolutionary algorithms; Metropolis-within-Gibbs; Global optimization.

1 Introduction

The inherent difficulty of global optimization problems lies in finding the very best minimum from a multitude of local minima. We consider the problem

of finding the global minimum of the unconstrained continuous optimization problem

$$\min f(x) \quad \text{such that} \quad x \in \Omega \subset \Re^n,$$

where $f(x)$ is a nonlinear function and x is a vector of continuous and bounded variables. A global minimization algorithm aims at finding the global minimizer x^* of $f(x)$ such that

$$f^* = f(x^*) \leq f(x), \quad \forall x \in \Omega.$$

Such an optimization problem arises in many practical fields of application, generally involving a large number of continuous variables, so there is a need for designing robust algorithms capable of solving problems with different characteristics within each field.

Random search is one of the pillars of most heuristic methods in global optimization. By introducing stochastic perturbations (e.g. the mutations in a genetic algorithm) it is possible to explore large regions of a landscape and potentially escape from local minima, resulting in the exploration of different local minima points. The optimal magnitude of this perturbation, in order to achieve a good balance between exploration and explotation, is a problem dependent task. In general, this dependency makes the parameters selection a major issue of heuristic algorithms design. In this paper we propose a Markov Chain Monte Carlo (MCMC) procedure that, in combination with a local search strategy, can find very competitive solutions to large global optimization problems in comparation with both deterministic and stochastic stablished methods. During execution, our algorithm adaptively determines adequate exploration and intensification rates. Due to the fact that the exploration stage is given by a clearly defined stochastic process, it is possible to have robust and meaningful control parameters.

In order to construct a global exploration strategy, the well known analogy between optimization problems and equilibrium in physical systems [14] is used. Consider a cost function $f(x_1, x_2, ..., x_n, ..., x_N)$. The probabilty density of a physical system at thermal equilibrium under the potential f is given by

$$p(\vec{x}) = \left(\frac{1}{Z}\right) exp(-f/kT),$$

where T is the temperature and k is the Boltzmann constant. At small values of the kT term, sampling from the equilibrium density will generate points close to the global optimum. However, if the kT term is too small, most of MCMC methods will suffer a large risk of getting trapped in local regions. This situation evidences the need for an accurate selection of the step size parameters, which dictate the amount of noise in the random search. A carefully tuned set of step size parameters for a given temperature may be not be appropriate for a different temperature. Moreover, a logarithmic schedule should be imposed to avoid premature convergence [11].

An attractive alternative to usual Metropolis-Hastings based approaches, as simulated annealing, is the use of Gibbs sampling. The main reason is that Gibbs sampling does not require the definition of any step size parameter and, in addition, the random search processes generated by it are capable of jumping large low probability regions [8]. Furthermore, convergence to the correct density is geometric under general conditions [5, 19]. Gibbs sampling, however, has the disadvantage that explicit expressions for the conditionals densities of interest are required. These conditionals can be provided only for particular density shapes. This drawback has been recently addressed by the Stationary Fokker-Planck (SFP) sampler, which generalizes the Gibbs sampler for arbitrary densities, at the cost of using some gradient information [3]. The SFP method has already been applied to global optimization as an exploration mechanism [3, 4]. Here, a similar approach is followed, but having as improvement the fact that no gradient information is required. Our method is based on the "Metropolis within Gibbs" (MG) algorithm proposed in [1]. The simplicity of MG makes it easy to define intensification strategies and adaptive simulated annealing type cooling schedules.

The rest of the paper is organized as follows. In the next section adaptive Gibbs sampling is introduced. Section 3 presents some background information on the class of estimation of distribution optimization algorithms, to which our proposed algorithm belongs. Section 4 presents the estimation of distribution algorithm with the adaptive Gibbs sampling proposed in this paper. Sections 5 and 6 introduce a number of test problems and several comparative methods, respectively. In Section 7 the algorithm is empirically evaluated and the results are analyzed, while emphasizing those aspects that are more difficult to tackle for any global or local optimization method, namely the increase of dimensionality and the presence of very rough landscapes. Concluding remarks are given in the section 8.

2 Adaptive Gibbs (AG) Sampling

As mentioned before, the proposed method is based on the "Metropolis within Gibbs" (MG) algorithm [1]. In the Gibbs sampler a Markov chain that converges to the density of interest $p(\vec{x})$ is constructed by sampling from the conditionals $p(x_n|\{x_{j\neq n}\})$. Simulating one value in turn for each individual variable from these conditionals is called one cycle of Gibbs sampling. Under general conditions, draws from this simulation algorithm will converge to the target density at a geometric rate [5, 19]. If it is not possible to directly sample from the conditionals, a solution is to incorporate a Metropolis type algorithm to simulate from each of them. These reasonings are the essential steps in the MG method. Following, candidate points for each variable are generated by

$$x_n^* = x_n^t + c_n Z, \tag{1}$$

where Z is a standard normal variate and c_n is a scale parameter. The candidate point will be accepted with probability

$$P = \min\left[1, p(x_1, x_2, ..., x_n^*, ..., x_N)/p(x_1, x_2, ..., x_n^t, ..., x_N)\right];$$

otherwise $x_n^{t+1} = x_n^t$. Therefore, at sufficiently large values of the c_n's the acceptance rates should be low, and as the c_n's tend to zero, the acceptance rates will tend to 1. This feature permits not only to define cooling shedules, but more important, to give criteria for the exploration of the landscape at the single variable level.

We have chosen cooling schedules of the form

$$c_n = c_n^o \tau^{-\alpha}, \alpha > 0, \tag{2}$$

where c_n^o is a constant chosen so that initally the acceptance rates are close to zero. The variable τ represents the actual iteration number. At each iteration a number of G Gibbs cycles are performed.

3 Estimation of Distribution Algorithms (EDAs)

Estimation of distribution algorithms are evolutionary algorithms that work with a population of candidate solutions (individuals). Initially, a random sample of individuals is generated. These individuals are evaluated using a cost function, which evaluates how accurate each solution is for the problem. Based on this evaluation, a subset of individuals is selected. Hence, individuals with better cost function values have a bigger chance of being selected. A probabilistic model for the selected solutions is then built, and a new set of individuals is sampled from the model. This process is iterated until the optimal solution has been found or another termination criterion is fulfilled. EDAs replace the genetic operators of crossover and mutation by estimating and sampling a probability distribution [16]. Moreover, EDAs differ in the way the information is gathered during the optimization process, and later use this information to build probabilistic models, which are used in turn to generate new solutions. Algorithm 1 shows the pseudo-code of the algorithmic structure behind most EDAs. EDAs are a promising tool for solving hard optimization problems in both discrete and continuous spaces. There has been a growing interest for EDAs in the last years. It is out of the scope of this paper to describe the approaches taken to implement the ideas we have just described. For a comprehensive introduction to the field see the works of [13, 16, 18].

4 Estimation of Distribution Algorithm with the Adaptive Gibbs Sampling (AGEDA)

Based on the EDAs initial approach, we propose some modifications that result in a new method for unconstrained global optimization, which we call estimation

Algorithm 1 General pseudocode framework for an EDA.

1: Given population size M.
2: Set $t \Leftarrow 0$. Generate $M \gg 0$ individuals at random.
3: **for** $t = 1, 2, ...$ until stopping criterion is met **do**
4: Evaluate individuals using the cost function.
5: Select $N < M$ individual according to selection methods.
6: Estimate the distribution $p^t(x)$ of the selected set.
7: Sampling M new individuals according to the distribution $p^t(x)$.
8: Set $t \Leftarrow t + 1$.
9: **end for**

of distribution algorithm with adaptive Gibbs sampling (AGEDA). In our approach, the way estimation and sampling are made is by means of the adaptive Gibbs sampling method described in section 2, where samples approximate the joint posterior distribution from the set of conditional posterior distributions. The fully joint probability distribution characterizes the problem being solved. Thus, we use adaptive Gibbs sampling to generate a set of new potential individuals and, in combination with a local search strategy, improves the individual solutions produced in each iteration of the algorithm. The adequate use of both the local information of solutions found and the global information about the search space improves the performance of our method. Algorithm 2 shows the pseudo-code of the proposed AGEDA.

In Algorithm 2, for each adaptive Gibbs sampling we also have acceptance rates (c_n's). These acceptance rates adaptively determine both the adequate exploration of the landscape at the single variable level and the intensification strategy to improve the solutions. In steps 8 to 14 we replace the value of the n-th variable ($x_n^{(best)}$) at random, if the acceptance criterion ($\varepsilon = 0.95$) is fulfilled. These steps conform the exploration part of the algorithm. In Steps 15 to 17 we improve the solution of the best individual $x_t^{(best)}$ via local search strategy, if the acceptance criterion ($\beta = 0.7$) is fulfilled. In $\langle c_n$'s$\rangle$, the brackets represent the average over the number of variables. We use the Nelder Mead method (described later) to improve the solution. These steps conform the intensification part of the algorithm. The parameters ε and β were empirically calibrated, so they are considered recommended parameters.

5 Test problems

In order to empirically evaluate the AGEDA algorithm, we selected some well known problems that act as performance tests for global optimization algorithms. These test problems were selected for testing the robustness of the AGEDA against stochastic and deterministic methods in three aspects that, even individually, decrease the performance of many global optimization algorithms, namely the increase in dimensionality, the multimodal function optimization and the optimization of nonsmooth functions. The selected test problems are

Algorithm 2 General pseudocode framework for the proposed AGEDA.

Given population size M and initial scale parameters c_n's.
Set $t \Leftarrow 0$. Generate $M \gg 0$ indiviuals at random.
Evaluate individuals using the cost function.
Select the best start individual $x_t^{(best)}$ for adaptive Gibbs sampling.
while termination criteria are not met **do**
  Generate M new individuals via adaptive Gibbs sampling (by using c_n's and $x_t^{(best)}$).
  Evaluate individuals using the cost function and select the best individual $x_t^{(best)}$.
  for $n = 1$ to N **do**
    if $c_n > \varepsilon$ **then**
      Update c_n by using equation 2.
      Replace the value of the n-th variable ($x_{n,t}^{(best)}$) at random.
    end if
    Update c_n by using equation 2.
  end for
  if $\langle c_n\text{'s}\rangle > \beta$ **then**
    Improve the solution of the best individual $x_t^{(best)}$ via local search strategy.
  end if
  Set $t \Leftarrow t + 1$.
end while

now introduced.

5.1 Rosenbrock problem

The Rosenbrock function is a well known test problem for optimization algorithms. Figure 1 exhibits the Rosenbrock function for two variables, where it can be seen that the global minimum is inside a long, narrow, parabolic shaped flat valley. Finding the valley is a trivial task, but converge to the global minimum is diffult. For this reason, it has been reported in the literature as a very difficult task for stochastic heuristics [12] and is very well suited to study the behavior of the algorithms while increasing the problem dimension. The problem is defined as follows,

$$\min \sum_{n=1}^{N} 100(x_{n+1} - x_n^2)^2 + (x_n - 1)^2,$$
$$-10 \leq x_n \leq 10,$$

and has the known global solution $x^* = (1, ..., 1)$, for which the cost function value is zero.

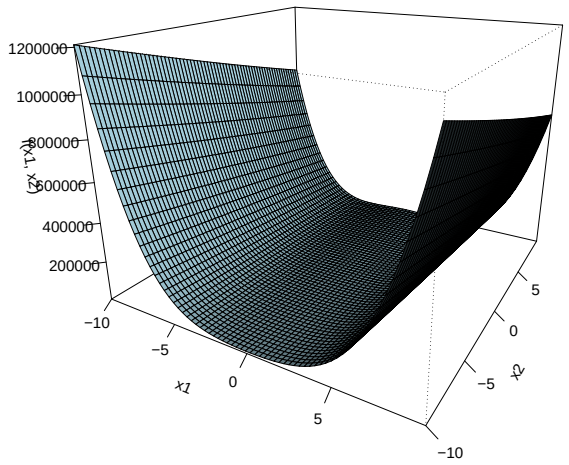

Figure 1 – The Rosenbrock function in 2D, $f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (x_1 - 1)^2$.

5.2 Morse clusters

An important application for global optimization techniques is the minimization of potential energy structures, which is relevant in the study of proteins and nanomaterials. The Morse potential is an adequate model for several atomic clusters and gives a challenging benchmark for global optimization algorithms [10]. The model consists in an expression for the pairwise atomic interactions,

$$V_{ij} = e^{2\rho(1-r_{ij})} - 2e^{\rho(1-r_{ij})},$$

where r_{ij} is the interatomic euclidean distance and ρ is a parameter that represents the equilibrium pair separation.

The problem is to minimize the potential energy of the N atom cluster,

$$V = \sum_{i<j} V_{ij}.$$

Fitting to bulk data indicates that, by the Morse model, realistic predictions can be made for clusters like C_60 (using $\rho = 13.62$), sodium (with $\rho = 3.15$) and nickel ($\rho = 3.96$), just to mention a few. The minimum energy configurations are of fundamental importance in addressing the chemical and physical properties of a given system.

5.3 Fractal function

One of the main interests in the development of heuristics is their use in problems for which an exact solution is not easily attainable. Functions with very rough landscapes are one of the most challenging problems for both exact and heuristic global optimization methods. Fractal function have strong similarities to real-world problems. Here, we consider at first instance a test problem with a fractal landscape introduced in [2],

$$\min f(x) = \sum_{n=1}^{N} C'(x_n) + x_n^2 - 1,$$
$$-5 \leq x_n \leq 5,$$

$$C'(x) = \begin{cases} \frac{C(x)}{C(1)|x|^{2-D}}, & \text{if } x \neq 0 \\ 1, & \text{if } x = 0 \end{cases}$$

$$C(x) = \sum_{j=-\infty}^{\infty} \frac{1 - \cos(b^j x)}{b^{(2-D)j}},$$

where $C(x)$ is an approximation of the Weierstrass-Mandelbrot cosine fractal function. For this function, D is known to be a box dimension ($1 < D < 2$) and it represents a parameter that arbitrarily increases or decreases the complexity of the cost function. For this fractal function it is impossible to indicate the exact position of the global minimum. Due to the zigzagging peaks close to the origin, several local optima with function values smaller than zero exist. Figure 2 introduces the fractal function with parameters $D = 1.85$ and $b = 1.5$. It can be seen in this figure the complexity of performing an optimization routine for this function, due to both the multiple locally optimal points within each region and the fact that gradient information cannot be used to determine the direction in which the function is decreasing. The $D = 1.85$ and $b = 1.5$ parameters are used in all the experiments within this paper.

6 Comparative Algorithms

Having selected suitable performance test problems for evaluating the AGEDA algorithm robustness, the task was then to select competitors for every single test problem. For achieving this we picked some methods, both deterministic and stochastic, that have been proved to reach good-quality solutions in at least one of the test problems. The methods we selected for the comparative tests with AGEDA are presented below.

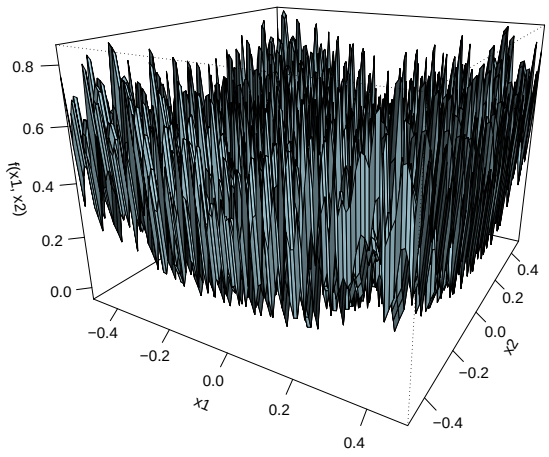

Figure 2 – Zoom on the fractal function in 2D with parameters D=1.85 and b=1.5.

6.1 Stochastic methods

In order to compare the proposed AGEDA with population-based heuristic search methods, we considered three of the most popular strategies nowadays, namely Genetic Algorithms (GAs), Particle Swarm Optimization (PSO) and Differential Evolution (DE). All three heuristics start with a set of randomly generated solutions (called individuals) which are updated throughout an iterative process using different mechanisms. GAs, PSO and DE methods have reported comparable results in highly complex problems such as some of the problems we used for evaluation in this work. Due to their proved performance, these techniques have been widely studied and, as a consecuence, there exist many versions of each of these strategies. To avoid biasing our study towards specific implementations, we consider the standard/basic versions of GAs, PSO and DE methods.

GAs: The genetic algorithms are inspired from biological evolution, where solutions (chromosomes) are coded as binary vectors and new individuals are created or updated by a recombination of selected individuals and mutation rules. In this work we considered[1] the canonical genetic algorithm with roulette wheel selection, one-point cross-over and a standard mutation procedure, see [9] for details.

PSO: A particle swarm optimization is inspired by the behavior of biological

[1]We used the Matlab® GAs toolbox implementation.

societies, such as flocks of birds and shoals of fishes, which present local and social behavior for achieving common goals [6]. Solutions are coded as numerical vectors (particles) and they are updated by combining information from global and local solutions that are found during the search process. For the comparison we implemented[2] the standard PSO algorithm with adaptive inertia weight, which is one of the most used improvements of PSO for enhancing the rate of convergence of the algorithm [21].

DE: The differential evolution is the newest population-based heuristic that we consider for the experimental comparison. In DE solutions are updated by combining existing solutions and adding randomness into this combination. The update of solutions is defined by simple rules for selection, cross-over and mutation. In this paper we considered[3] the basic DE algorithm as described in [22].

6.2 Deterministic Methods

In addition to stochastic methods, we considered two classical deterministic algorithms that have shown their capabilities to achieve good-quality solutions when implementing them to solve large optimization problems, namely, the Nelder-Mead (NMM) method and the Conjugate Gradients Algorithm (CGA).

NMM: The Nelder Mead method is a simplex method for finding a local minimum of a function of several variables that has been devised by Nelder and Mead [17]. The NMM requires only function evaluations, not derivatives. In the N-dimensional space, a simplex is a polyhedron with $N + 1$ points (or vertices). We chose the $N+1$ points and defined an initial simplex. The method iteratively updates the worst point by four operations: reflection, expansion, one-dimensional contraction and multiple contraction. The NMM uses a small number of function evaluations per iteration and it is one of the most widely used direct search methods for multidimensional nonlinear optimization problems that have a unique optimal solution. A big disadvantage of the NMM is that it can converge to non-stationary points [15]. For the experimental comparison we used the package *neldermead* available in the CRAN packages repositories[4].

CGA: The conjugate gradient method is an algorithm for finding the nearest local minimum of a function of n variables which presupposes that the gradient of the function can be computed. It uses conjugate directions instead of the local gradient for going downhill [7]. The CGA combines the information from all previous directions in such a way as to create a subsequent search direction that is independent (or conjugated) to all previous directions. For the experimental comparison we used the package *Rcgmin* available in the CRAN packages repositories.

[2] The source code is available upon request.

[3] We used the DE implementation available at: http://www.icsi.berkeley.edu/∼storn/code.html

[4] http://cran.r-project.org/

7 Experimental results

In this section we evaluate the AGEDA performance to solve several well known test problems for unconstrained global optimization. These test problems were selected for testing the robustness of the AGEDA against stochastic and deterministic methods in three aspects that individually decrease the performance of many global optimization algorithms, namely the increase in dimensionality, the multimodal function optimization and the optimization of nonsmooth functions. The empirical evaluation consisted in assessing the performance of the methods based on five independent executions of each method and for each problem size, from 5 to 55 variables. Both the cost function average value and the average number of cost function evaluations performed are reported for each problem dimension. The algorithm that reached the best cost function average value for each problem dimension is marked in bold face. Every test problem implementation have its own main stopping criterion, but in order to make the comparisons as fair as possible, we have set an additional stopping criterion consisting in the number of cost function evaluations (FEs), which is set to 500,000 for all experiments in this paper. Furthermore, we make use of statistics to validate the results of our empirical evaluation and we present a ranking for comparing the overall performance of each algorithm when applying them to the selected test problems.

7.1 Rosenbrock problem

Given that the optimal cost function value for the Rosenbrock problem is zero, we consider as the main stopping criterion reaching a cost function value of 0.001 or lower. Defining such threshold for acceptable solutions is necessary as we are dealing with heuristic search methods that do not guarantee obtaining the global optimum (at least for a finite number of iterations).

Tables 1 and 2 display the cost function average value ($f(\hat{x})$ column) and the average number of cost function evaluations needed (# of FEs column) by every algorithm when applied to the Rosenbrock problem while varying the problem dimensionality. The standard deviation for both measures is reported as well (Std. Dev. columns). It is noteworthy that all tables within this paper follow this format.

It can be seen that, for all the dimensions, the method with the best performance is the CGA, as it finds objective function values close to the global optimum while using a few cost function evaluations. Note that the number of cost function evaluations for the CGA method is an estimate of the number of evaluations necessary to compute the gradient and the hessian throughout the process. For the dimensions 5 to 30 (Table 1), the AGEDA method achieves solutions close to zero. For the rest of the dimensions, we observed that AGEDA needs more than 500,000 cost function evaluations to achieve high-quality solutions. Nevertheless, the results obtained by AGEDA are the closest, in comparation to the rest of the comparative methods, to the results achieved by the CGA method. The behavior of the NMM method is acceptable for problems

of low dimensionality (5-20), although its performance drops significantly for dimensions larger than 20. Among the stochastic optimization techniques, the DE technique obtained the best results for the dimensionalities of 5-30, followed by PSO and GA; however, the performance of DE is rather poor for dimensionalities larger than 30. PSO and GA methods showed a more stable behavior across dimensionalities.

Table 1 – Experimental results when applying the selected comparative algorithms to the Rosenbrock problem for 5 to 30 dimensions. All results have been averaged over five independent runs.

N	method	$f(\hat{x})$	Std. Dev.	# of FEs	Std. Dev.
5	AGEDA	2.34E-14	5.18E-14	1000	0
	CGA	**9.19E-19**	1.38E-18	357	85
	NMM	0.0086	0.0010	587	196
	PSO	0.0053	0.0035	487720	27458
	DE	0.0007	0.0001	99221	10052
	GA	0.5774	0.9092	500000	0
10	AGEDA	1.92E-8	3.69E-8	10200	3834
	CGA	**4.04E-18**	5.66E-18	506	116
	NMM	0.0095	0.0000	4551	2173
	PSO	0.6176	0.3120	500000	0
	DE	0.0008	0.0001	339241	7759
	GA	2.23	3.06	500000	0
15	AGEDA	7.19E-6	1.11E-5	41062	9293
	CGA	**4.77E-18**	8.03E-18	749	93
	NMM	0.0095	0.0000	31345	12606
	PSO	3.22	3.10	488480	25759
	DE	0.1542	0.0382	500000	0
	GA	18.54	31.12	500000	0
20	AGEDA	7.88E-5	9.53E-5	122518	17303
	CGA	**3.13E-18**	6.66E-18	941	83
	NMM	0.0099	0.0000	120235	69874
	PSO	7.63	6.60	500000	0
	DE	7.16	0.2096	500000	0
	GA	47.29	39.86	500000	0
25	AGEDA	1.56E-4	1.18E-4	278530	35925
	CGA	**4.25E-18**	6.01E-18	990	134
	NMM	15.78	10.032	205129	146579
	PSO	14.75	8.17	500000	0
	DE	15.89	0.3082	500000	0
	GA	55.04	38.80	500000	0
30	AGEDA	3.24E-4	1.68E-4	496824	2310
	CGA	**1.29E-17**	1.07E-17	1258	260
	NMM	57.28	57.50	447232	391524
	PSO	32.47	25.45	500000	0
	DE	25.53	1.08	500000	0
	GA	73.36	34.10	500000	0

Table 2 – Experimental results when applying the selected comparative algorithms to the Rosenbrock problem for 35 to 55 dimensions. All results have been averaged over five independent runs.

N	method	$f(\hat{x})$	Std. Dev.	# of FEs	Std. Dev.
35	AGEDA	3.21	2.35	496986	1299
	CGA	**7.28E-18**	1.13E-17	1480	201
	NMM	102.43	78.78	346808	313588
	PSO	56.86	31.47	500000	0
	DE	187.79	109.75	500000	0
	GA	80.86	36.26	500000	0
40	AGEDA	10.21	2.92	497846	2657
	CGA	**1.52E-18**	1.38E-18	1550	254
	NMM	192.18	163.19	457433	211480
	PSO	65.24	28.46	500000	0
	DE	1026.58	322.72	500000	0
	GA	128.29	52.28	500000	0
45	AGEDA	19.19	6.54	498498	1402
	CGA	**2.44E-18**	2.49E-18	1706	326
	NMM	366.91	457.39	433512	91419
	PSO	73.24	30.97	500000	0
	DE	8864.54	1418.65	500000	0
	GA	155.11	43.83	500000	0
50	AGEDA	31.65	1.04	497710	1817
	CGA	**3.35E-18**	3.26E-18	1863	376
	NMM	142.66	85.10	500000	0
	PSO	83.71	35.76	500000	0
	DE	64437.43	15089.31	500000	0
	GA	152.85	36.89	500000	0
55	AGEDA	42.79	1.57	497479	1712
	CGA	**2.76E-18**	4.18E-18	2084	264
	NMM	218.82	139.86	495701	9613
	PSO	119.70	34.74	500000	0
	DE	348102.44	60328.55	500000	0
	GA	164.71	24.30	500000	0

7.2 Morse clusters

For the Morse clusters problem we allow the methods under evaluation to run until either their own default stopping criterion is met (related to a number of successive iterations with no improvement in the solution) or the maximum number of function evaluations is reached. For the comparisons we have set the following error function

$$error = \frac{|f(x^*) - f(\hat{x})|}{|f(x^*)|} < 0.1, \tag{3}$$

where $f(x^*)$ is the putative known global optima and $f(\hat{x})$ is the average values of the objective function.

The putative global optima for each of the considered clusters are given in Tables 3 and 4. Despite the fact that very effective heuristics for Morse cluster

optimization exist [10], they are particularly designed for this problem; while our interest in this paper is on the development of a general purpose method.

Tables 3 and 4 show both the average value of the cost function and the average number of cost function evaluations needed by every algorithm when applyed to the Morse clusters problem while varying the problem dimensionality. It is important to point out that N denotes the number of atoms to clusterize in a three-dimensional space, so the real problem size is $S = 3N$, being S the overall number of variables to optimize. From our experiments, it can be seen that for dimensions $N = 15$, 20 and 25 the method with the best performance is the CGA, while for the rest of dimensions this distinction belongs to the NMM method. It should be noted that, for large dimensions (30-55), the solutions achieved by AGEDA are closer to the solutions provided by the best performance method than the rest of comparative algorithms, while for low dimensions (5-10) most of the comparative methods seem to achieve solutions with almost equal quality. Note that, for dimension $N = 50$, AGEDA does not reach the criterion of equation 3 with 500,000 cost function evaluations. The NMM does not reach either the criterion of equation 3 for $N = 55$. For $N = 40$ to 55 the quality of the solution of the CGA method diverges. Regarding the stochastic techniques, the three methods obtained lower performances for dimensionalities larger than 15.

7.3 Fractal function

When solving the nonsmooth fractal function, we have adopted similar stopping criteria as those considered in the Morse cluster problem: the method stops either when its own default stopping criterion is met or when the maximum number of function evaluations is reached.

Tables 5 and 6 report the average of the cost function values and the number of cost function evaluations needed by every algorithm when applyed to the nonsmooth fractal function problem while varying the problem dimensionality. It can be seen that for all dimensions in this problem the best performance is obtained by the GA method, reaching cost function values smaller than zero while evaluating the cost function 500,000 times. For all dimensions, the AGEDA method achieves cost function values smaller than zero. The AGEDA solutions are very comparable to those of GA method in all dimensions. The behavior of the DE method is similar to the behavior of both the GA and the AGEDA methods for low dimensions (5-30), although its performance drops significantly for dimensions larger than 30, where the AGEDA is the only method, among the others, that remains close to the performance of the most effective method.

7.4 Ranking comparison

In order to illustrate the overall performance of each algorithm with respect to the others, we have constructed a simple ranking by taking into consideration the information shown in Tables 1 to 6. The best function value has a rank 1, the second best function value has a rank 2 and so on, until the worst method

Table 3 – Experimental results when applying the selected comparative algorithms to the Morse cluster problem for 5 to 30 dimensions. All results have been averaged over five independent runs.

N	method	$f(x^*)$	$f(\hat{x})$	Std. Dev.	# of FEs	Std. Dev.	*error*
	AGEDA		**-9.04**	0.0000	3603	226	**0.0000**
	CGA		**-9.04**	0.0000	3125	326	**0.0000**
5	NMM	-9.04	**-9.04**	0.0000	5156	2700	**0.0000**
	PSO		**-9.04**	0.0000	500000	0	**0.0000**
	DE		-7.40	0.3665	500000	0	0.1817
	GA		**-9.04**	1.94E-5	500000	0	**0.0000**
	AGEDA		-26.05	0.9114	14801	4025	0.0516
	CGA		-25.35	0.4757	81216	42044	0.07718
10	NMM	-27.47	**-26.53**	0.7860	22429	6487	**0.0342**
	PSO		-24.91	3.14	500000	0	0.0934
	DE		-7.97	1.49	500000	0	0.7100
	GA		-25.51	1.13	500000	0	0.0716
	AGEDA		-45.98	0.8293	55829	28601	0.0757
	CGA		**-46.67**	1.79	197148	57527	**0.0617**
15	NMM	-49.75	-46.45	1.86	149418	49892	0.0663
	PSO		-30.99	6.16	500000	0	0.3770
	DE		-9.40	2.29	500000	0	0.8110
	GA		-37.20	5.44	500000	0	0.2520
	AGEDA		-66.55	1.14	63203	12458	0.0821
	CGA		**-68.21**	1.99	361000	111125	**0.0592**
20	NMM	-72.51	-66.47	1.24	301985	44931	0.0831
	PSO		-35.78	3.23	500000	0	0.5070
	DE		-8.44	1.31	500000	0	0.8840
	GA		-45.09	4.13	500000	0	0.3780
	AGEDA		-86.34	0.5009	79769	20075	0.0923
	CGA		**-88.96**	2.68	811800	312243	**0.0648**
25	NMM	-95.13	-88.81	1.46	500000	0	0.0664
	PSO		-48.48	5.55	500000	0	0.4900
	DE		-9.41	0.87	500000	0	0.9010
	GA		-55.00	5.20	500000	0	0.4220
	AGEDA		-107.18	0.4270	184306	61085	0.0949
	CGA		-86.71	9.05	877830	497173	0.2678
30	NMM	-118.43	**-109.23**	3.50	500000	0	**0.0777**
	PSO		-52.41	5.08	500000	0	0.5570
	DE		-10.36	1.57	500000	0	0.9130
	GA		-65.39	5.13	500000	0	0.4480

has a rank 6. Tables 7 to 10 show the rank information for the Rosenbrock, the Morse clusters and the fractal problem, respectively. In each one of these Tables, the rows exhibit the comparative methods while the columns are related to the problem dimension. On this form, each element of the Table from columns 5 to 55 corresponds to the rank given to a specific method for that dimension, according to its performance when compared to the other methods. The lower the rank, the better the performance of the method. Column R-Sum shows the sum of the ranks per method. Finally, the R-Rank column introduces the rank of

Table 4 – Experimental results when applying the selected comparative algorithms to the Morse cluster problem for 35 to 55 dimensions. All results have been averaged over five independent runs.

N	method	$f(x^*)$	$f(\hat{x})$	Std. Dev.	# of FEs	Std. Dev.	*error*
	AGEDA		-128.73	0.7365	172786	58764	0.0931
	CGA		-22.12	22.77	772548	248565	0.8441
35	NMM	-141.96	**-133.61**	1.62	500000	0	**0.0588**
	PSO		-43.19	12.72	500000	0	0.6960
	DE		-8.98	2.55	500000	0	0.9370
	GA		-73.75	7.35	500000	0	0.4800
	AGEDA		-151.63	0.1280	311787	73343	0.0973
	CGA		147.58	30.59	970632	44211	1.87
40	NMM	-167.99	**-155.30**	2.31	500000	0	**0.0755**
	PSO		-47.08	6.91	500000	0	0.7200
	DE		-10.79	3.63	500000	0	0.9360
	GA		-70.18	7.09	500000	0	0.5820
	AGEDA		-174.57	0.2903	490773	8739	0.0952
	CGA		574.15	27.94	1001880	187917	3.97
45	NMM	-192.95	**-178.21**	3.866	500000	0	**0.0764**
	PSO		-38.76	5.93	500000	0	0.7990
	DE		-12.23	4.73	500000	0	0.9370
	GA		-74.55	9.42	500000	0	0.6140
	AGEDA		-183.89	3.50	488098	7098	0.1634
	CGA		1281.06	43.65	1289680	403702	6.82
50	NMM	-219.82	**-199.23**	5.81	500000	0	**0.0936**
	PSO		-48.91	11.01	500000	0	0.7780
	DE		-9.73	1.81	500000	0	0.9560
	GA		-78.26	4.58	500000	0	0.6440
	AGEDA		-194.68	13.61	485588	151	0.2221
	CGA		2417.50	47.75	1485000	792375	10.65
55	NMM	-250.29	**-222.10**	6.54	500000	0	**0.1126**
	PSO		-47.02	6.59	500000	0	0.8120
	DE		-10.14	0.5690	500000	0	0.9590
	GA		-69.63	6.08	500000	0	0.7220

the R-Sum column, which is an indicative of the overall performance for a given method in all dimensions for the solved problem. The overall performance is introduced in Table 10. The column R-Sum exhibits the sum of ranks obtained in columns R-Rank for all the problems. The Rank column is the overall rank, among all test problems. From this column, we can see that AGEDA obtained the best rank across the different problems and dimensions, fact that illustrates the robustness of our method while increasing the dimensionality, and when optimizing multimodal and nonsmooth functions. Despite the fact that other methods, such as CGA, outperformed the AGEDA in one problem, AGEDA achieved consistently a good performance across the different tasks.

Table 5 – Experimental results when applying the selected comparative algorithms to the nonsmooth fractal function problem for 5 to 30 dimensions. All results have been averaged over five independent runs.

N	method	$f(\hat{x})$	Std. Dev.	# of FEs	Std. Dev.
5	AGEDA	-0.4299	0.0264	69442	150
	CGA	40.32	16.06	60	109
	NMM	-0.0960	0.2180	807	92
	PSO	25.19	2.92	500000	0
	DE	-0.0875	0.0832	500000	0
	GA	**-0.9713**	0.1213	500000	0
10	AGEDA	-0.2353	0.2612	137692	687
	CGA	79.41	19.32	13	1
	NMM	3.65	7.47	6017	5958
	PSO	57.19	2.05	500000	0
	DE	-0.1056	0.0902	500000	0
	GA	**-1.71**	0.1295	500000	0
15	AGEDA	-0.5459	0.0798	203339	1329
	CGA	123.38	21.60	19	14
	NMM	6.39	7.13	8620	4080
	PSO	90.02	2.26	500000	0
	DE	-0.0996	0.0493	500000	0
	GA	**-2.25**	0.1075	500000	0
20	AGEDA	-0.2637	0.2448	268635	165
	CGA	172.50	41.13	22	16
	NMM	4.28	4.70	17317	3963
	PSO	120.39	2.26	500000	0
	DE	-0.0884	0.0171	500000	0
	GA	**-2.86**	0.0956	500000	0
25	AGEDA	-0.1368	0.4908	409620	1832
	CGA	213.67	27.81	33	34
	NMM	5.69	3.72	23968	4114
	PSO	156.86	4.76	500000	0
	DE	-0.1161	0.1161	500000	0
	GA	**-3.34**	0.0855	500000	0
30	AGEDA	-0.3233	0.4788	495054	2661
	CGA	308.77	68.85	14	3
	NMM	8.61	6.37	95174	33072
	PSO	190.31	2.04	500000	0
	DE	-0.1032	0.1154	500000	0
	GA	**-4.07**	0.2200	500000	0

7.5 Statistical comparison

In order to statistically validate the robustness of AGEDA, we use the Welch's t test (or unequal variance t test). The Welch's t test is a technique used to compare means of two samples when cannot be safely assumed that population variances are equal [20]. On the one hand, by using this test we could infer if the difference between two means (average cost function values in this case) are significantly different, and so they come from different populations. Such a result will indicate that two methods are not likely to produce equal quality

Table 6 – Experimental results when applying the selected comparative algorithms to the nonsmooth fractal function problem for 35 to 55 dimensions. All results have been averaged over five independent runs.

N	method	$f(\hat{x})$	Std. Dev.	# of FEs	Std. Dev.
35	AGEDA	-0.2227	0.9916	496146	42
	CGA	287.59	47.65	26	29
	NMM	16.17	12.53	204497	123646
	PSO	230.14	8.19	500000	0
	DE	0.0483	0.2103	500000	0
	GA	**-4.07**	0.2200	500000	0
40	AGEDA	-0.2947	0.7748	496689	1275
	CGA	326.16	31.02	41	31
	NMM	20.93	6.58	237184	195298
	PSO	259.32	3.52	500000	0
	DE	0.8034	0.0691	500000	0
	GA	**-4.29**	0.1956	500000	0
45	AGEDA	-1.83	0.0549	497508	86
	CGA	388.51	67.05	12	1
	NMM	13.03	8.95	335779	272687
	PSO	295.04	2.74	500000	0
	DE	1.83	0.3261	500000	0
	GA	**-5.01**	0.1499	500000	0
50	AGEDA	-0.1885	0.3486	495735	1350
	CGA	412.74	29.15	11	1
	NMM	18.07	9.23	300187	211061
	PSO	326.04	4.74	500000	0
	DE	3.15	0.3563	500000	0
	GA	**-5.26**	0.1978	500000	0
55	AGEDA	-0.6541	0.9725	495544	926
	CGA	453.84	36.21	17	13
	NMM	25.05	8.57	500000	0
	PSO	368.25	4.96	500000	0
	DE	8.76	0.5502	500000	0
	GA	**-5.22**	0.2319	500000	0

Table 7 – Ranking results for the Rosenbrock problem.

Method/N	5	10	15	20	25	30	35	40	45	50	55	R-Sum	R-Rank
AGEDA	2	2	2	2	2	2	2	2	2	2	2	22	2
CGA	1	1	1	1	1	1	1	1	1	1	1	11	1
NMM	5	4	3	3	4	5	5	5	5	4	5	48	4
PSO	4	5	5	5	3	4	3	3	3	3	3	41	3
DE	3	3	4	4	5	3	6	6	6	6	6	52	5
GA	6	6	6	6	6	6	4	4	4	5	4	57	6

solutions. On the other hand, we could construct and compare the means confidence intervals to infer how close are the solutions qualities of two methods. The Welch's t test is performed under the assumption of normality. We use the samples (the five independent runs) of the solutions obtained by the three best methods for each problem of the highest dimension ($N = 55$) for the statistical

Table 8 – Ranking results for the Morse problem.

Method/N	5	10	15	20	25	30	35	40	45	50	55	R-Sum	R-Rank
AGEDA	1	2	3	2	3	2	2	2	2	2	2	23	2
CGA	1	4	1	1	1	3	5	6	6	6	6	40	4
NMM	1	1	2	3	2	1	1	1	1	1	1	15	1
PSO	1	5	5	5	5	5	4	4	4	4	4	46	5
DE	2	6	6	6	6	6	6	5	5	5	5	58	6
GA	1	3	4	4	4	4	3	3	3	3	3	35	3

Table 9 – Ranking results for the Fractal problem.

Method/N	5	10	15	20	25	30	35	40	45	50	55	R-Sum	R-Rank
AGEDA	2	2	2	2	2	2	2	2	2	2	2	24	2
CGA	6	6	6	6	6	6	6	6	6	6	6	72	6
NMM	3	4	4	4	4	4	4	4	4	4	4	46	4
PSO	5	5	5	5	5	5	5	5	5	5	5	60	5
DE	4	3	3	3	3	3	3	3	3	3	3	38	3
GA	1	1	1	1	1	1	1	1	1	1	1	12	1

Table 10 – The final ranking results.

Method	R-Sum	Rank
AGEDA	6	1
NMM	9	2
GA	10	3
CGA	11	4
PSO	13	5
DE	14	6

tests. GNU R was used for this statistical significance study.

The normality of the samples distribution is checked by using the Shapiro-Wilk test. Table 11 shows both the W statistic and the p-values computed. The null hypothesis for this test is that the data is normally distributed. The Rosenbrock problem is significant with a level of significance $\alpha = 0.01$ (W critical = 0.6859). The Morse and Fractal problem are significant with a level of significance $\alpha = 0.05$ (W critical = 0.7620). One would accept the null hypothesis, concluding that there is no information to discard normality in the data.

Table 12 shows the statistic t, the p-values and the confidence interval computed for all pairwise comparisons concerning AGEDA when applying the Welch's t test. The null hypothesis is that the two population means are the same, but the two population variances may differ. It can be seen, for all comparisons, that the resulting p-values obtained in this test clearly indicate statistically significant differences between every two methods, that is, all results are significant at the 5% significance level. One would reject the null hypothesis, concluding that there is strong evidence that the expected values for all pairwise comparisons are different, which means that the three methods offer different

quality solutions. For a further analysis, we make use of the confidence interval. In the rest of this section, when we refer to the difference between means, we stand for the difference between means predicted by the confidence interval.

For the Rosenbrock problem, the difference between the means of CGA (best method) against AGEDA seems to be smaller than the difference between the means of PSO (2nd best method) against AGEDA. Moreover, for the Morse cluster problem, the difference between the means of NMM (best method)against AGEDA is much smaller than the difference between the means of GA (2nd best) against AGEDA. Finally, for the Fractal problem, the difference between the means of GA (best method) against AGEDA is smaller than the difference between the means of DE (2nd best) against AGEDA. These results indicate that, on average, AGEDA's solutions are the closest to the solutions provided by the best performance algorithm in each of the three test problems at the highest dimension. None of the other algorithms share this property. Therefore, we conclude that there is strong evidence indicating that AGEDA is a robust approach for difficult high dimensional unconstrained global optimization tasks.

Table 11 – The Shapiro-Wilk normality test results.

Function	Method	W	p-value
Rosenbrock	CG	0.7362	0.0220
	AGEDA	0.7790	0.0540
	PSO	0.8893	0.3534
Morse	NMM	0.8849	0.3320
	AGEDA	0.8603	0.2293
	GA	0.9232	0.5508
Fractal	GA	0.9075	0.4525
	AGEDA	0.8741	0.2834
	DE	0.9361	0.6384

Table 12 – Welch's t test results.

Function	Comparison	Statistic t	p-value	Confidence interval
Rosenbrock	AGEDA versus CGA	60.77	4.38E-07	[40.82, 44.73]
	PSO versus AGEDA	4.94	0.0077	[33.80, 120.03]
Morse	AGEDA versus NMM	4.05	0.0072	[10.71, 44.11]
	GA versus AGEDA	18.75	3.18E-06	[108.40, 141.70]
Fractal	AGEDA versus GA	10.22	0.0002	[3.38, 5.76]
	DE versus AGEDA	18.85	8.75E-07	[8.21, 10.62]

8 Conclusions

In this paper we have proposed and presented a new approach in the field of estimation of distribution algorithms (EDAs) based on adaptive Gibbs sampling

for unconstrained global optimization, called AGEDA. AGEDA directly extracts global statistical information about the search space during the search and, in combination with a local search strategy (local information), can find very competitive solutions to large global optimization problems, when comparing with deterministic and stochastic stablished methods. The adequate use of both the local information of solutions found and the global information about the search space improves the performance of the proposed method. The adaptive Gibbs sampling adaptively determines the adequate exploration of the landscape at the single variable level, as well as the intensification rates for this task.

We have evaluated the performance of our method against deterministic and stochastic algorithms that are commonly employed for solving challenging well known test problems. For the selection of the test problems, we have focussed in problems that involve three central aspects that mainly determine the difficulty of global optimization problems, namely high-dimensionality, multi-modality and non-smoothness. For the selection of the comparative algorithms, we have considered three of the most popular heuristic strategies nowadays, namely Genetic Algorithms (GAs), Particle Swarm Optimization (PSO) and Differential Evolution (DE), and two classical deterministic algorithms that have shown their capabilities to achieve good-quality solutions when implementing them to solve large optimization problems, namely, the Nelder-Mead (NMM) method and the Conjugate Gradients Algorithm (CGA).

Experimental results show that our approach is statistically robust, as it is capable of finding reasonable quality solutions for problems involving a high dimensionality, a nonsmooth function and a function with multiple local optimas, which are three critical aspects that define the difficulty of a global optimization problem. By robust we refer to the fact that, on average, AGEDA performs better than other methods in high dimensional problems, as it achieves high quality solutions for all the test problems.

In the future, we will extend theses performance comparisons against other methods in the state-of-the-art for unconstrained global optimization, and we will extend the AGEDA approach to execute in parallel computing. For the AGEDA parallel approach, we will study problems with larger dimensions than provided here, which conform a more challenger test for our method and other heuristics.

Acknowledgements

This work was partially supported by the National Council of Science and Technology of Mexico (CONACyT), under grant 206705 and by UANL-PAICYT program under grant "Inference based on density estimation".

References

[1] Albert J.: *Bayesian Computation with R.* Springer, 2007.

[2] Bäck T.: *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms.* Oxford University Press, New York, 1996.

[3] Berrones, A.: Stationary probability density of stochastic search processes in global optimization. *J. Stat. Mech.*, P01013, 2008.

[4] Berrones, A.: Bayesian Inference Based on Stationary FokkerPlanck Sampling. *Neural Computation*, 22(6), 1573-1596, 2010.

[5] Canty, A.: Hypothesis tests of convergence in markov chain monte carlo. *Journal of Computational and Graphical Statistics*, 8, 93-108, 1999.

[6] Engelbrecht A. P.: *Fundamentals of Computational Swarm Intelligence.* Wiley, 2006.

[7] Fletcher R. and Reeves C.: Function minimization by conjugate gradients. *The Computer Journal*, 7 (2): 149-154, 1964.

[8] Geman S. and Geman D.: Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721-741, 1984.

[9] Goldberg D. E.: *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, 1989.

[10] Pinter D. J.: *Global Optimization: Scientific and Engineering Case Studies* Springer, 2006.

[11] Kirkpatrick, S., Gelatt, Jr., C. D. and Vecchi M. P.: Optimization by Simulated Annealing. *Science*, 220:671-680, 1983.

[12] Laguna M. and Martí R.: Experimental testing of advanced scatter search designs for global optimization of multimodal functions. *Journal of Global Optimization*, 33(2):235-255, 2004.

[13] Larrañaga P. and Lozano J.: *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation.* Genetic Algorithms and Evolutionary Computation, Vol. 2. Springer, 2001.

[14] Metropolis N., Rosenbluth A., Rosenbluth M., Teller A., Teller E.: Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087-1092, 1953.

[15] Mckinnon K. I. M.: Convergence of the Nelder-Mead simplex method to a nonstationary point. *SIAM Journal on Optimization*, 9, 148-158, 1999.

[16] Mühlenbein H. and Paaß G.: From recombination of genes to the estimation of distributions I. Binary parameters. *In Lecture Notes in Computer Science 1411: Parallel Problem Solving from Nature-PPSN IV*, 178-187, 1996.

[17] Nelder J. A. and Mead R.: A simplex method for function minimization, *The Computer Journal*, 7(2):308-313, 1965.

[18] Pelikan M., Goldberg D. E. and Lobo F.: A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications*, 21(1):520, 2002.

[19] Roberts G. O. and Polson N. G.: On the Geometric Convergence of the Gibbs Sampler. *J. R. Statist. Soc. B, 56*, 377-384, 1994.

[20] Sawilowsky and Shlomo S.: Fermat, Schubert, Einstein, and BehrensFisher: The Probable Difference Between Two Means When $\sigma_1 \neq \sigma_2$. *Journal of Modern Applied Statistical Methods*, 1(2), 2002.

[21] Shi Y. and Eberhart R. C.: Emprirical study of particle swarm optimization. *IEEE In Proceedings of the Congress on Evolutionary Computation*, Piscataway, NJ, USA, 19451949, 1999.

[22] Storn R. and Price K.: Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11:341359, 1997.